

Juuso Lakka

TIEDONHALLINTAJÄRJESTELMÄ SÄHKÖSUUNNITTELUUN

Opinnäytetyö
Sähkötekniikan koulutusohjelma


Helmikuu 2013




MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

KUVAILEHTI

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Opinnäytetyön päivämäärä	
Tekijä(t) Juuso Lakka		Koulutusohjelma ja suuntautuminen Sähkötekniikan koulutusohjelma	
Nimeke Tiedonhallintajärjestelmä sähkösuunnitteluun			
Tiivistelmä <p>Opinnäytetyöni sai alkunsa kehitystarpeesta, jonka tarkoituksena oli tuoda tehokkuutta sähkösuunnitteluun. Sovelluksen tavoitteena on keskittää suunnittelu yhteen sovellukseen sekä parantaa projektien, tiedostojen ja piirien hallittavuutta ja laatua. Sovelluksen avulla täytyi pystyä muokkaamaan tietokannassa sijaitsevia tietoja yksinkertaisesti ja tehokkaasti, mihin Microsoft Access ei soveltunut, joka Etteplan Design Centerillä aiemmin oli käytössä.</p> <p>Sovellus luotiin Visual Studio 2010:n avulla käyttäen Visual Basic –kieltä .NET Frameworkissä. Raportissani keron, kuinka ohjelma on luotu, kuinka tietoa käsitellään erillisen rajapinnan kautta sekä kuinka tieto kulkee tietokannan ja rajapinnan välillä.</p> <p>Sovelluksen tekeminen alkoi sovelluksen ominaisuuksien suunnittelemisesta ja ohjelmoinnin avulla päädyttiin lopputuloksiin, joihin Etteplan Design Center ja minä olimme tyytyväisiä. Sovellus toimii hyvänä mallina tehokkuuden parantamisesta ja tulevien kehitystöiden perustana.</p>			
Asiasanat (avainsanat) SQL, Visual Basic.NET, Suunnittelujärjestelmä, tiedonhallinta			
Sivumäärä 27 s. + liitt. 1 s.	Kieli Suomi	URN	
Huomautus (huomautukset liitteistä)			
Ohjaavan opettajan nimi Hannu Honkanen		Opinnäytetyön toimeksiantaja Etteplan Design Center Oyj, Kouvola	

DESCRIPTION

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Date of the bachelor's thesis 	
Author(s) Juuso Lakka		Degree programme and option Electrical Engineering	
Name of the bachelor's thesis Data management system for designing of electrical systems.			
Abstract <p>The aim of this Bachelor's thesis was to develop an application to concentrate and improve the quality of the design job and make projects, files and circuits more manageable. The main priority of the application is to manage the database efficiently and rapidly, which was impossible by using Microsoft Access.</p> <p>The application was created by using Visual Studio 2010 and Visual Basic language in .NET Framework. This report discusses how the application was created, how the data is handled with from external interface and how data transfers between interface and database. Data management, tools and programming languages which were used in my work, are also discussed in this thesis.</p> <p>The application concentrates and improves quality of the design as were set earlier in the goals and in that way it also increases productivity of designers work.</p>			
Subject headings, (keywords) SQL, Visual Basic.NET, Data management			
Pages 27 p. + app. 1 p.		Language Finnish	
URN 			
Remarks, notes on appendices 			
Tutor Hannu Honkanen		Bachelor's thesis assigned by Etteplan Design Center Oyj, Kouvola	

SISÄLTÖ

1	JOHDANTO	1
2	ETTEPLAN DESIGN CENTER	2
3	TIEDONHALLINTA	3
3.1	Relaatiotietokanta.....	3
3.2	Tietokannanhallintajärjestelmät.....	4
3.3	Tietokantojen käyttö sähkösuunnittelussa.....	4
4	SÄHKÖPIIRIT	6
4.1	Keskuskojeet.....	7
4.2	Kaapelit.....	9
4.3	Kenttälaitteet.....	10
4.4	Kuormat.....	11
4.5	Järjestelmän liityntä	11
5	OHJELMOINTI JA TYÖVÄLINEET	12
5.1	Visual Studio	12
5.2	Microsoft .NET Framework	12
5.3	SQL	13
5.4	Crystal Reports.....	13
5.5	AutoCAD.....	14
6	SUUNNITTELUJÄRJESTELMÄ.....	14
6.1	EttePro	14
6.2	Elsa	15
6.3	Tietojen käsittely	18
6.4	Raportointi	19
6.5	Tiedostojen tulostaminen.....	20
7	VERTAILU	22
7.1	Ohjelmiston kehittyminen	22
7.2	Taloudellinen näkökulma	22
8	POHDINTA	25
	LÄHTEET	26

LIITE

1 Kaapeliluettelo

1 JOHDANTO

Finanssivetoistuneessa ja jatkuvasti kehittyvässä maailmassa asioita halutaan tehdä nopeammin, ja työskentelytapoja on jatkuvasti kehitettävä, jotta pysytään globaalissa kilpailussa mukana. Kehittämisen avulla saadaan asiakkaalle tarjottua palveluita edullisempaan ja kilpailullisempaan hintaan.

Suunnittelupalvelujen tehostamiseksi on siirrytty käyttämään tietokonepohjaisia suunnittelutyökaluja, joiden avulla saadaan kuvia piirrettyä nopeammin, sekä kuvia on helpompi muokata jälkikäteen. Tietokoneiden kehittyminen on mahdollistanut yhä raskaampien ja suurempien kokonaisuuksien hallinnan helposti. Suurille kokonaisuuksille on jouduttu luomaan uusia palveluita, jolloin on siirrytty työpöytäsovelluksista pilvipalveluihin, joita voidaan käyttää mistä vain.

Opinnäytetyöni tarkoituksena on luoda sovellus, jonka avulla saadaan sähkösuunnittelua tehokkaammaksi. Sovelluksen tulee toimia Etteplan Design Center Oyj:n nykyisen suunnittelujärjestelmän rinnalla, eli sovelluksen pitää pystyä muokkaamaan suunnittelujärjestelmän käyttämää tietokantaa. Uuden sovelluksen avulla työskentelystä koetaan ottaa manuaalisia vaiheita pois, joissa työntekijän tarvitsee syöttää tietoja, jotka voitaisiin tehdä tietokoneenkin avulla ja vähentää tietojen syöttämistä kahteen kertaan. Sähkösuunnittelussa kuvien tulisi olla virheettömiä, joten sovelluksen avulla inhimillisten virheiden määrä yritetään minimoida.

2 ETTEPLAN DESIGN CENTER

Etteplan on teollisten laitteistojen suunnitteluun ja teknisen tuoteinformaation ratkaisuihin ja palveluihin erikoistunut asiantuntijayritys. Vuonna 2011 Etteplanin liikevaihto oli 119,4 miljoonaa euroa. Yhtiöllä on tällä hetkellä yli 1 800 asiantuntijaa Suomessa, Ruotsissa, Hollannissa ja Kiinassa. Etteplan on listattu NASDAQ OMX Helsinki Oy:ssä tunnuksella ETT1V.

Etteplanin strategian tavoitteena on kasvattaa yhtiön arvoa hyödyntämällä liiketoimintaympäristössä tapahtuvia muutoksia ja sen tarjoamia mahdollisuuksia. Strategian neljä kulmakiveä ovat asiakaskeskeisyys, palveluratkaisut, suunnittelumenetelmät ja yksi Etteplan. /1./

Etteplanin arvoihin kuuluu asiakastyytyväisyys, henkilöstön hyvinvointi ja ammattitaitoinen toimintatapa /1/.

Etteplan Design Center Oy:n Kouvolan toimisto on erityisesti puunjalostusteollisuutta sekä kone- ja laitetoimittajia palveleva insinööritoimisto. Toimisto on perustettu vuonna 1993 nimellä LCA Engineering Oy ja vuodesta 2007 se ollut osa Etteplan Oyj -konsernia. Kouvolan toimipiste tarjoaa erilaisia ratkaisuja prosessi-, tehdas-, kone-, sähkö-, automaatio-, LVI-, energia-aloilla ja työllistää yhteensä noin 60 henkilöä. /2./

3 TIEDONHALLINTA

Työssäni jouduin tutustumaan tiedonhallintaan liittyviin käsitteisiin, kuten tietokantoihin ja tiedonhallintajärjestelmiin.

”Tiedon varastoinnin tavoitteena on helpottaa tietojenkäsittelyä; varastoidut tiedot kuvaavat tietojenkäsittelyn kohteena olevaa ”todellisuutta”, ne toimivat näin ihmisen muistin jatkeena ja vähentävät tietojenkäsittelyssä tarvittavien tietojen tarpeetonta uudelleenkeräämistä” /3, s. 5/.

Tietojen lisääntyminen tarkoittaa myös tietojenhallintajärjestelmien jatkuvaa kehittymistarvetta. Sähköalalla tietojen lisääntyminen johtuu siirtymisestä paperidokumentaatiosta sähköiseen dokumentaatioon sekä sähköpiirimäärien jatkuvasta kasvamisesta.

3.1 Relaatietietokanta

”Tietokanta on kokoelma tiettyä kohdetta kuvaavia tietoja, joita yksi tai useampi tietosysteemi käyttää ja päivittää” /3, s. 63/.

Etteplanin suunnittelujärjestelmä käyttää JET –relaatietietokantaa, koska sen ylläpito on helppoa Microsoft Accessin kautta. Microsoft Accessin valinta oli viisasta, koska käyttäjämäärä ei ole suuri, ja Microsoft Accessia käytettäessä investoinnin tuotto-prosenti on usein korkea /4, s. 12/.

”Relaatietietokantamallissa järjestelmä hallitsee kaikkia tietoja taulukoissa. Yhteen Taulukkoon tallennetaan tiedot yhdestä aihepiiristä, mikä sisältää sarakkeita, joissa on erityyppisiä tietoja aiheesta sekä rivejä, jotka kuvaavat yhden aiheeseen liittyvän tapausten ominaisuudet” /5, s. 5/.

Relaatietietokantamallin hyödyntäminen vähentää datan määrää, kun samaa tietoa ei tarvitse syöttää kahta kertaa, vaan tarvittavat tiedot voidaan hakea taulusta, johon on viitattu. Esimerkiksi asiakkaan tiedot voidaan syöttää yhteen tauluun, johon viitataan tilaukset -taulusta, täten tilaustauluun ei tarvitse syöttää asiakkaan tietoja. Relaatietie-

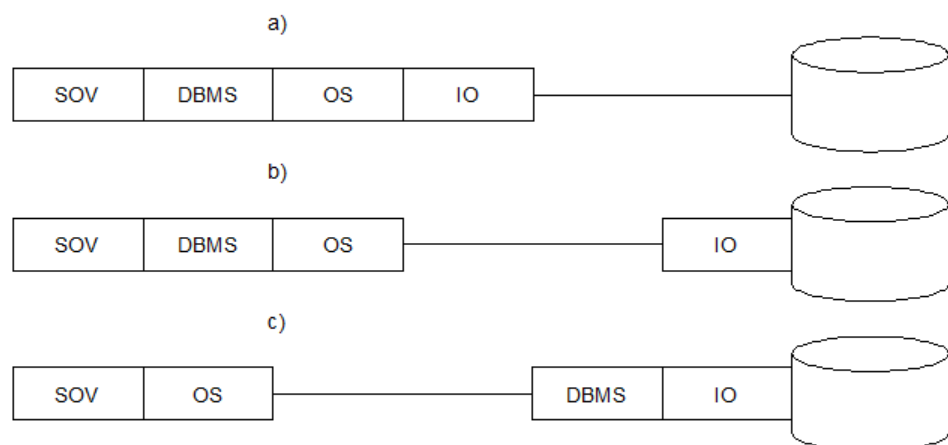
tokanta soveltuu sähköpiireille hyvin, koska tiedot ovat usein toisiinsa liittyviä ja hierarkkisia.

3.2 Tietokannanhallintajärjestelmät

Tietokantojen kasvaessa suuriksi ja haluttaessa käsitellä tietoja helposti ja tehokkaasti, tarvitaan tietokannanhallintajärjestelmä. (DBMS = Database management system)

Monesti sanotaan käytettävän Microsoft Access –tietokantoja, vaikka todellisuudessa Microsoft Access toimii tietokannanhallintajärjestelmänä, jolla hallitaan JET –tietokantaa.

Tietokantakäsittelyn tapoja on monia, kuten kuvasta 1 voidaan huomata:



KUVA 1. Tietokantakäsittelyn eri tapoja /3, s. 71/

Sovellukseni toimii perinteisellä tietokantakäsittelyn tavalla (kuva 1, a): sovellusohjelma (SOV) suoritetaan samalla tietokoneella kuin tietokannan hallintajärjestelmää (DBMS), jotka toimivat käyttöjärjestelmän (OS) alaisuudessa, joka valvoo syöttöä ja tulostusta (I/O) /3, s. 70/.

3.3 Tietokantojen käyttö sähkösuunnittelussa

Tietokantojen käyttö sähkösuunnittelussa on yleistynyt, kun tietokoneavusteinen suunnittelu (CAD) on yleistynyt. Ennen paperille piirretyt dokumentit luodaan nykyään tietokoneen avulla.

CAD –kuvilla sijoitetut objektit sijaitsevat tietokannassa. Eli käytännössä jokainen CAD –piirros on tietokanta, josta ohjelma hakee objektien tiedot ja näin osaa sijoittaa ne kuvaan oikeaan kohtaan.

Tuotetut dokumentit on myös säilöttävä jonnekin, mistä ne ovat helposti löydettävissä ja järjestettävissä, mikä on tehnyt dokumenttienhallintajärjestelmistä yleisimpiä. Yhtenä esimerkkinä dokumenttienhallintajärjestelmistä on M-Files.

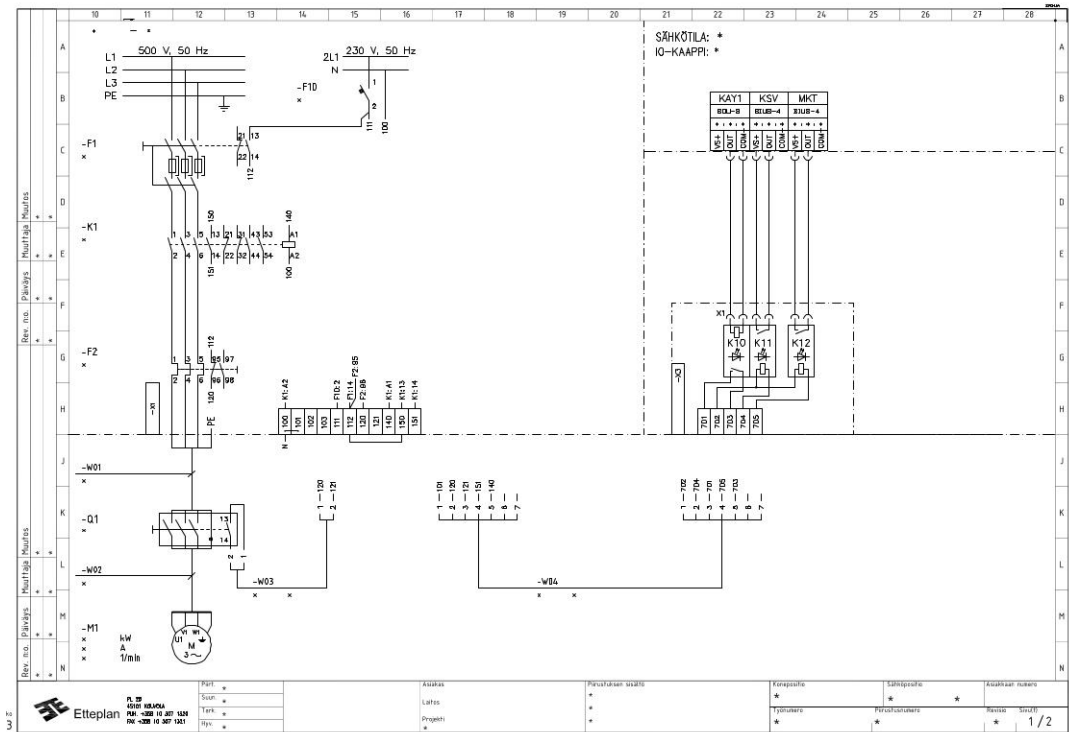
Sähkösuunnitelmien ohella tarvitsee myös tilaajalle luovuttaa muita dokumentteja, joiden tiedot on kerätty sähkösuunnitelmista, kuten kaapeliluetteloita, kojeluetteloita ja kilpiluetteloita. Tietojen ollessa syötettynä tietokantaan voidaan samat tiedot saada helposti useaan dokumenttiin, jolloin tietoja ei tarvitse syöttää useaan kertaan, mikä nopeuttaa prosessia ja vähentää inhimillisten virheiden määrää.

”Sähkösuunnittelutyötä tarvitaan jossain muodossa aina, kun laite, järjestelmä, kone rakennus tai kiinteistö tarvitsevat sähköenergiaa toimiakseen” /6, s. 9/.

Sähköpiiri on usein laitekohtainen, eli jokainen sähkökeskuksen lähtö luokitellaan omaksi piiriksi. Sähköpiiri sisältää uniikin keskuslähdön, sähköposition ja koneposition. Sähköpiiri koostuu keskuksessa sijaitsevista kojeista, kentälle menevistä kaapeleista, kentällä sijaitsevista moottoreista ja kentälaitteista sekä järjestelmään menevistä sisään- ja ulostuloista.

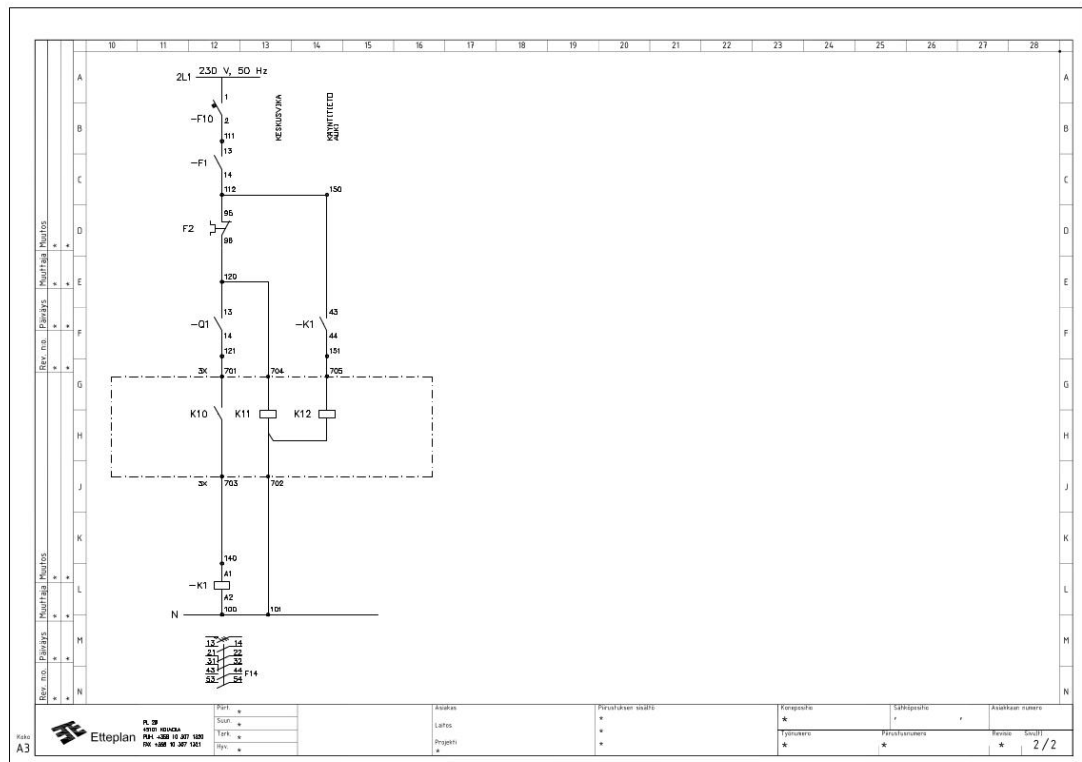
Jokaiselle sähköpiirille luodaan omat dokumentit, jonka sivumäärät vaihtelevat yhdestä sivusta lukuisiin sivuihin. Sähköpiirien dokumentit koostuvat yleensä piirikaaviosta ja johdotuskaavioista.

Johdotuskaaviossa esitetään sähköpiirien osien sijainti ja kytkennät. Johdotuskaavion perusteella asentajan on helppo tehdä esitetyt kytkennät.



KUVA 2. Piirin johdotuskaaviosta

Piirikaaviossa esitetään piirin toiminta. Vian paikallistaminen piirikaavion avulla on helpompaa.



KUVA 3. Piirin piirikaaviosta

Saman sähköpiirin johdotus- ja piirikaavio sisältävät samat asiat, minkä vuoksi piiri- ja johdotuskaavioiden tulee olla yhdenmukaiset.

4.1 Keskuskojeet

Yleensä sähkötilassa sijaitseva kenno- tai kotelokeskus on jaettu lähtöyksikköihin, missä keskuskojeet sijaitsevat. Kuvassa 4 voidaan nähdä kuva kennokeskuksesta ja kuvassa 5 voidaan nähdä kuva kotelokeskuksesta.



KUVA 4. Kennokeskus /7/



KUVA 5. Kotelokeskus

Keskuskojeita ovat esimerkiksi varokkeet, lämpöreleet, kontaktorit, moottorisuojareleet ja kytkimet.



KUVA 6. Lähtöyksikkö, kuvassa nähtävissä kytkinvaroke (1.), kaksi kontaktoria (2.) ja lämpörele (3.)

Keskuskojeiden tehtävänä on ohjata laitteen toimintaa, suojata kentällä sijaitsevia komponentteja, erottaa laite sähköverkosta ja estää sähkönjakeluverkkoa kaatumasta.

Keskuskojeet mitoitetaan piirin vaatiman kuorman ja kuorman tyyppin perusteella.

4.2 Kaapelit

Kaapeleiden avulla siirretään sähkötehoa tai tietoa laitteiden välillä. Kaapeleiden ominaisuudet riippuvat käyttöpaikoista ja –tarkoituksista.

Voimakaapeleiden avulla siirretään sähkötehoa kuormalle, kuten moottorille. Kuorman vaatima jännite vaihtelee yleensä 230-690V välillä, sekä tehot voivat olla jopa megawatteja, minkä vuoksi yhtälön 1 perusteella virrat voivat nousta erittäin suuriksi.

$$P = UI \quad (1)$$

Suurten virtojen takia johdin lämpenee, jos johtimen poikkipinta-ala ei ole tarpeeksi suuri. Tämän vuoksi voimansiirrossa käytettävien kaapeleiden poikkipinta-alat ovat suurempia kuin tiedonsiirrossa. Teollisuudessa yleisimpänä kaapelilajina voimansiirrossa käytetään MCMK:ta (kuparivoimakaapeli) ja AMCMK:ta (alumiinivoimakaapeli). Voimakaapelit mitoitetetaan keskuskojeiden ja kuorman perusteella.

Tiedonsiirrossa käytettävissä kaapeleissa siirretään vain binääri- tai analogiatietoa, joiden vaatimat tehot eivät ole suuria, joten kaapeleiden poikkipinta-alat ovat pienemmät. Jännitetasoilla 24V ja 48V siirrettäessä kaapeleina käytetään yleensä JAMAK- ja NOMAK –instrumentointikaapeleita. 230V jännitteellä käytetään kaapeleina yleensä MMO –ohjauskaapelia, jonka poikkipinta-ala on suurempi kuin instrumentointikaapeleilla, yleensä 1,5mm².

4.3 Kenttälaitteet

Kenttälaitteet sijaitsevat kentällä, eli keskuksen ja moottorin välillä. Kenttälaitteita ovat taajuusmuuttajat, erilaiset vahdit ja kytkimet, kuten yleisimpänä turvakytkin.

Kenttälaitteiden sijoittelu vaihtelee: taajuusmuuttaja sijoitetaan usein samaan sähkötilaan, kuin missä keskus sijaitsee, jolloin se on suojassa ympäristön rasitteilta. Monien valmistajien taajuusmuuttajan ympäristön lämpötila saa olla -15°C – 40°C, mikä prosessitiloissa voi alittua tai ylittyä.

Turvakytkimen tarkoituksena on suojata käyttäjää sekä estää laitteen odottamaton käynnistyminen huoltotyön aikana. Turvakytkin sijoitetaan usein mahdollisimman lähelle moottoria.

4.4 Kuormat

Sähköpiirin tarkoitus on jakaa kuorman tarvitsema teho ja ohjaus. Kuorman tyyppi riippuu käyttötarkoituksesta, erilaisia käyttötarpeita ovat lämmitys ja mekaaninen energia, kuten pumput ja puhaltimet.

Kuorman tyyppi vaikuttaa tarvittavaan suojaukseen. Esimerkiksi moottorit täytyy suojata lämpöreleillä, jotka valvovat moottorin kuormitusta mittaamalla johtimissa kulkevaa virtaa.

Teollisuudessa ulkona sijaitseviin putkiin täytyy asentaa saattolämmitys, jotta neste putkissa ei jäätyisi. Lämmitykset täytyy suojata vikavirtasuojilla eristysvikoja vastaan. Vikavirtasuojaa käytetään myös suojaamaan hengenvaaralliselta sähköiskulta. ”Jos resistiiviset eristysviat voivat aiheuttaa tulipalon. esim. kattolämmityselementeissä, vikavirtasuojan mitoitusvirta $I_{\Delta n}$ saa olla korkeintaan 30 mA” /8, s.153/.

4.5 Järjestelmän liityntä

Sähköpiireistä halutaan usein viedä tietoja järjestelmään, jossa mikroprosessorit voivat suorittaa haluttuja toimintoja tiedon perusteella. Sähköpiireistä yleisimpiä järjestelmään vietäviä tietoja ovat käyntitieto, erilaiset vikatiedot ja mahdollisen taajuusmuuttajan nopeustieto. Järjestelmästä voidaan myös ohjata piirin toimintaa.

Järjestelmän ja piirin välinen tiedonsiirto on analogista tai binääristä tietoa. Analogisilla tiedoilla saadaan esimerkiksi venttiilin asentotieto tai moottorin pyörimisnopeus, sekä analogisella viestillä voidaan ohjata moottorin nopeutta. Binääritiedoilla voidaan ohjata piirin toimintaa tai saada tarvittavia tietoja piiriltä järjestelmään.

Tiedot otetaan usein lähtöyksikön jostakin koskettimesta tai taajuusmuuttajasta, ja tieto välitetään kaapelia pitkin ristikytkentäkaapin riviliittimille, mitkä ovat johdotettu instrumenttikaapin korteille. Korteja on erilaisia: binääri input, binääri output, analogia input ja analogia output. Korttityypit muuttuvat riippuen valmistajasta ja jännite-
tasosta.

5 OHJELMOINTI JA TYÖVÄLINEET

Tässä kappaleessa käsitellään ohjelmointikieliä sekä työvälineitä, joiden avulla ohjelma saatiin valmiiksi.

5.1 Visual Studio

Visual Studio on Microsoftin kehittämä ohjelmointiympäristö, jonka avulla voidaan kehittää web –sovelluksia, perinteisiä työpöytäsovelluksia ja useita muun tyyppisiä sovelluksia /9/. Visual Studion ensimmäinen versio julkaistiin vuonna 95 nimellä Visual Studio, jonka jälkeen uudempia versioita on julkaistu muutaman vuoden välein lisätyillä ominaisuuksilla. Itse olen käyttänyt Visual Studio 2005- ja Visual Studio 2010-versioita projektini aikana.

Visual Studio tukee useita eri ohjelmointikieliä, kuten C++, C#, F#, VB.NET, jota minä käytin /10/. Kaikkia kieliä ei tueta suoraan, vaan joudutaan asentamaan erillisiä kielipaketteja.

Visual Studiosta löytyy myös eri versioita, joista löytyy erilaisia ominaisuuksia. Versioiden nimet ovat Express, Professional, Premium ja Ultimate. Itse käytin projektini luomisessa Professional versiota, joka Etteplanille hankittiin.

Express edition on ilmainen. Express edition on suunnattu opiskelijoille ja harrastelijoille, ja siitä on riisuttu ominaisuuksia /11; 12/. Professional Edition on maksullinen, siitä löytyy tuet kaikille kielille sekä muita ominaisuuksia /9/. Premiumista ja Ultimatesta löytyy vielä laajempi kokoelma ominaisuuksia.

5.2 Microsoft .NET Framework

.NET Framework on Microsoftin kehittämä kehitysympäristö. .NET Framework –kehitysympäristö tukee monia kieliä, ja se koostuu kolmesta pääosasta. CLR eli Common Language Runtime, joka on ohjelmien ajoympäristö. CLR vastaa myös muistin jakamisesta ja prosessien ja säikeiden käynnistämisestä ja pysäyttämisestä, mikä helpottaa ohjelmoijan työtä. Kehitysympäristö koostuu yhtenäisistä, oliopohjaisista, hierarkkisista ja laajennettavista luokkakirjastoista. Kehitysympäristön kolmas

pääosa on ASP.Net, jonka avulla voidaan helposti kirjoittaa ASP –web-sovelluksia. /12./

.NET Framework –kehitysympäristöstä löytyy valmiita kirjastoja, joiden tarkoituksena on nopeuttaa ja helpottaa sovelluskehitystä. Uusin versio .NET Frameworkista on 4.5, mutta itse käytin projektini aikana versiota 4.0.

”Visual Basic .NET (VB.NET) on .NET-ympäristöön tarkoitettu oliopohjainen versio Microsoftin Visual Basic –ohjelmointikielestä” /13, s.12/.

Valitsin ohjelmointikieleksi VB.NET:n, koska olin jo aiemmin käyttänyt hieman Visual Basic for Application –ohjelmointikieltä.

5.3 SQL

Ohjelma käyttää SQL –kieltä hakiessaan tietoja tietokannasta.

SQL eli Structured Query Language on tavallisin kieli, jonka avulla tietokantoihin päästään käsiksi ja niitä voidaan hallita. SQL on ANSI (American National Standards Institute) standardoitu, mutta SQL -kielestä on silti monia erilaisia versioita, mutta ne kaikki tukevat yleisimpiä komentoja, kuten SELECT, UPDATE, DELETE, INSERT ja WHERE. /14./

SQL:n avulla voidaan suorittaa monia toimintoja tietokannassa, kuten suorittaa kyselyitä, poistaa tietueita, luoda tauluja ja päivittää tietueita. Ohessa yksinkertainen kysely, joka hakee kaikki tiedot taulusta nimeltä ”Piirit”, missä tietueen Keskus –kolumnin arvo on ”SL01”.

```
SELECT * FROM Piirit WHERE Keskus = "SL01"
```

5.4 Crystal Reports

Crystal Reports on SAP:n tarjoama sovellus, jolla voidaan luoda raportteja. Crystal Reports tuli Visual Studion 2005 mukana, minkä vuoksi valitsin sen tähän projektiin. Visual Studio 2010:ssa Crystal Reportsia ei sisällytetty mukana, mutta se oli ladatta-

vissa siihen erillisenä lisäosana. Lisäosan asentaakseen täytyi omistaa vähintään Professional –versio Visual Studiosta.

5.5 AutoCAD

AutoCAD on Autodeskin vuonna 1982 kehittämä CAD (Computer Aided Design/Drafting) –sovellus, joka on yksi johtavista CAD –sovelluksista /15; 16/.

AutoLISP (List Processing) on AutoCADin tukema ohjelmointikieli. Automatisoimalla usein toistuvia tehtäviä AutoLISPin avulla voidaan lisätä tuottavuutta. AutoLISP –ohjelmointikieltä tuetaan vain AutoCADissa, ja sitä ei löydy AutoCAD LT:stä. /17, s. 1073./

AutoCAD 2013 –version mukana tulee sovellus nimeltä Autocad 2013 Core Console (accorreconsole), joka on komentorivi –versio AutoCADistä, ja jonka avulla piirustusten sarjaprosessointi nopeutuu. Accorreconsole on kevyempi sovellus kuin AutoCAD, mutta se sisältää lähes kaikki samat ominaisuudet kuin AutoCAD 2013 lukuunottamatta graafista käyttöliittymää (GUI). AccoreConsole tukee myös AutoLISP -komentoja. /16./

6 SUUNNITTELUJÄRJESTELMÄ

Suunnittelutyön tehostamiseksi käytetään tietokoneavusteista suunnittelua, mihin lue-
taan myös suunnittelujärjestelmien käyttö, jonka avulla suuria tietomääriä hallitaan ja
suoritetaan toimintoja, jotka on voitu automatisoida.

Suunnittelujärjestelmien käyttäminen on yleistynyt tietokoneiden kehittyessä. Suun-
nittelujärjestelmiä on markkinoilla monia, kuten Alma Software Oy:n ALMA, Sie-
mensin Comos ja Integraphin Integraph.

6.1 EttePro

Etteplanilla on käytössä Toivo Blombergin kehittämä suunnittelujärjestelmä: EttePro. EttePron avulla voidaan siirtää tietoa tietokannan eri tauluista .dwg-, .drw-, ja .dgn –
kuviin. EttePron käyttöliittymä on web-pohjainen.

EttePron käyttämät tietokannat sijaitsevat verkkolevyllä, jonne kaikilla työntekijöillä on oikeudet päästä. Tietokantojen muokkaaminen on tapahtunut käyttämällä Microsoft Access –tietokannanhallintajärjestelmää sen yksinkertaisen käyttöliittymän ja edullisuuden vuoksi, koska se tulee Microsoft Office –paketin mukana.

Tietokannassa sijaitsee taulut:

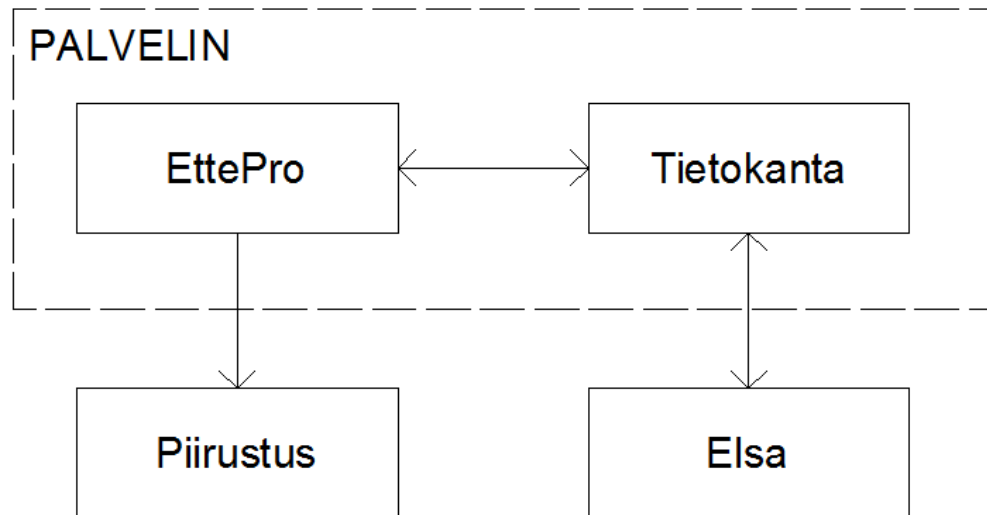
- Piirit–taulu, johon tallennetaan piirien tiedot.
- Kojeet–taulu, johon tallennetaan kojeiden tiedot.
- Kaapelit–taulu, johon tallennetaan kaapeleiden tiedot.
- Kenttälaitteet–taulu, johon tallennetaan kenttälaitteiden tiedot.
- Moottorit–taulu, johon tallennetaan kuormien tiedot.
- I/O–taulu, johon tallennetaan järjestelmään menevät I/O:t

Tietojen muokkaaminen Microsoft Accessin kautta on melko vaivalloista, koska joudutaan avaamaan useita tauluja, joihin syötetään eri tietoja. Monen taulun ollessa auki tulee helposti vahingossa syötettyä tietoja väärään tauluun tai väärälle piirille.

EttePron Web –käyttöliittymä on luotu lähinnä tarkastelemaan piirin tietoja ja generoimaan kuvia, koska käyttöliittymää kehitettäessä haluttiin, että muokkaaminen tapahtuu Microsoft Accessin kautta.

6.2 Elsa

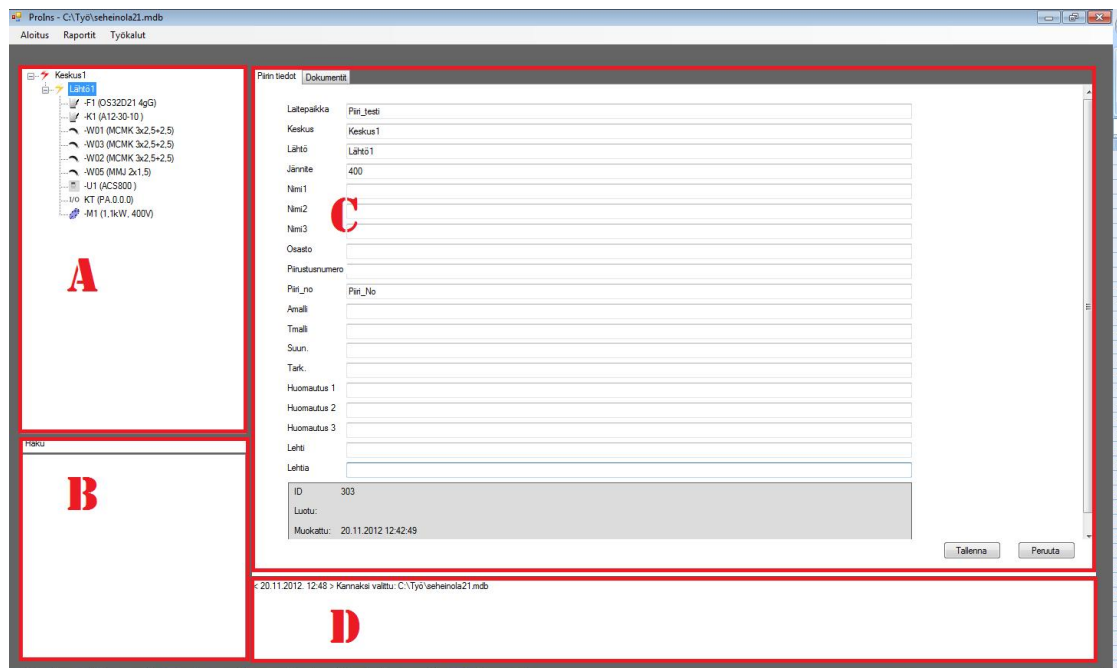
Nimesin luomani suunnittelujärjestelmän Elsaksi. Elsan tarkoituksena on toimia EttePron rinnalla työkaluna, jolla tietokannan sisältöä voidaan hallita sekä suorittaa erilaisia toimintoja, jotka helpottavat suunnittelijan tehtäviä.



KUVA 7. Kaavio suunnittelujärjestelmien riippuvuuksista

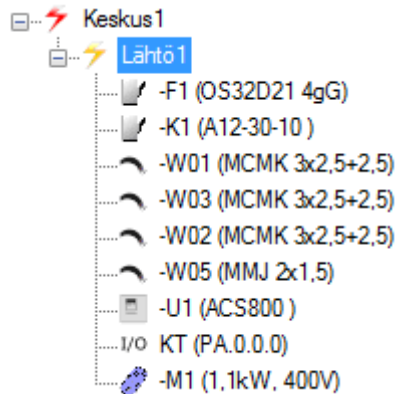
Elsa toimii työpöytäsovelluksena, eli se täytyy asentaa jokaiselle tietokoneelle erikseen. Elsa toimii vain Windows –käyttöjärjestelmässä.

Elsa koostuu useasta ikkunasta, mutta suurimmaksi osaksi käytetään päänäkymä –ikkunaa. Päänäkymä koostuu 2 puunäkymästä, paneelistä ja konsolirivistä.



KUVA 8. Päänäkymä (A. Puunäkymä 1, B. Puunäkymä 2, C. Paneeli, D. Konsoli)

Ylemmässä puunäkymässä (kuva 8, kohta a) näkyy piirit hierarkkisesti: ylimmällä tasolla keskusket, sitten lähtöyksiköt (piiri), jonka alla sijaitsee kojeet, kaapelit, kentälaitteet, kuormat ja I/O:t.



KUVA 9. Puunäkymän hierarkia

Paneeliin (kuva 8, kohta b) tulee tiedot valitusta objektista.

Konsoliin (kuva 8, kohta c) tulee ilmoituksia käyttäjälle.

Käynnistettäessä sovellus avautuu aloitusnäky, jossa näkyy suosikkikannat ja puumerkit, jos ne ovat asetettu käyttäjäkohtaisiin asetuksiin. Suosikkikannat ovat hyperlinkkejä, jolloin haluttu kanta saadaan sovelluksella nopeasti auki.

Asetuksissa käyttäjä pääsee asettamaan henkilökohtaisia asetuksia, joita ovat viisi suosikkikantaa ja puumerkit. Puumerkkien avulla tekijän nimi saadaan aseteltua automaattisesti piiriä lisätessä ja raportteja tulostettaessa.

Tietokannan saa myös avattua erillisellä ”Avaa tietokanta” –komennolla, jolloin voidaan valita mikä tahansa tietokanta. Ohjelmalla voidaan avata mikä tahansa tietokanta, jonka pääte on ”.mdb”, mutta kantaa avatessa tulostaa ohjelma konsoliriville virhesanoman, jos tietokannan rakenne ei ole oikea.

6.3 Tietojen käsittely

Elsan avulla tietokannasta voidaan tarkastella, muokata, lisätä, poistaa ja kopioida tietoja. Tietojen muokkaaminen tietokannassa tapahtuu erillisten SQL-lauseiden avulla. SQL -lauseita ei toteuteta ohjelman sisällä vaan tietokannassa säilöttyinä proseduureina (Stored Procedures).

Elsa käyttää Stored Procedureja, koska suoritettavat SQL-käskyt pysyvät samoina, jolloin dynaamisia SQL -käskyjä ei tarvita. Dynaamisten SQL-käskyjen ongelmana on tietoturva: SQL -käskyille voidaan syöttää helposti merkkejä, jotka aiheuttavat ohjelman kaatumisen, eli tietokanta on haavoittuvainen SQL Injection -tietoturvahyökkäyksille. Stored Procedureilla ei ole tätä ongelmaa, koska mahdolliset muuttujat syötetään parametreina. Tietoturva tässä sovelluksessa ei ole olennainen, koska tietokantaan kaikilla on täydet oikeudet, mutta ohjelman ei tule kaatua syötettäessä vääränlaisia merkkejä. Microsoft Access ei tue tarkalleen Stored Procedureja, mutta Stored Queryjä, joita Microsoft Accesissa voidaan luoda, voidaan käyttää samalla tavalla kuten Stored Procedureja.

Tietojen näyttäminen tapahtuu painamalla puunäkymässä objektia, jolloin objektin tiedot näytetään paneelilla. Paneelin näkymä ja suoritettava SQL-lause riippuu valitun objektin tyypistä. Paneelilla sijaitseviin tekstiruutuihin tulee tiedot piiristä, joita voidaan tarvittaessa muokata ja tallentaa erillisellä ”tallenna”-painikkeella. Tallennettaessa tiedot päivitetään tietokantaan, sekä objektin teksti muutetaan puunäkymään.

Poistettaessa tietoja valitaan objektit, jotka halutaan poistaa, minkä jälkeen avautuu uusi ikkuna, jossa kysytään käyttäjältä, haluaako hän varmasti poistaa objektit. Varmistuksen avulla vähennetään riskiä, että poistettaisiin vahingossa objekteja tai poistetaan väärinä objekteja. Objektia poistettaessa poistetaan objektin tiedot ja objektin alla hierarkiassa sijaitsevat tiedot tietokannasta sekä puunäkymästä.

Tietojen kopioiminen tapahtuu painamalla piiriä ja valitsemalla ”Kopio piiri”, jolloin avautuu ikkuna, jonka avulla voidaan lisätä uusi piiri tietokantaan. Kopioitaessa ikkunan tekstiruutuihin haetaan valitun piirin tiedot tietokannasta, jolloin käyttäjä pääsee muokkaamaan tietoja ennen uuden piirin luomista.

Uutta objektia luotaessa lisätään tiedot tietokantaan ja luodaan tarvittavat objektit puunäkymään.

Elsa toimii myös piirikohtaisten dokumenttien hallintajärjestelmänä. Piirikohtaiset dokumenttilinkit helpottavat suunnittelijan tehtäviä, koska piirin dokumenttilinkkeihin voidaan lisätä esimerkiksi valokuvia tai muita piiriin liittyviä dokumentteja. Näin dokumentit ovat helposti löydettävissä, eikä tarvitse selailla palvelimen hakemistoja löytääkseen tietyt dokumentit.

Piirin tietoja tarkastellessa voidaan paneelin toiselle sivulle lisätä linkkejä tiedostoihin. Tiedoston lisääminen tapahtuu raahaamalla tiedosto taulukon päälle, jolloin se lisätään tietokantaan. Tiedoston avaaminen tapahtuu kaksoisklikkaamalla taulukossa halutun tiedoston riviä.

6.4 Raportointi

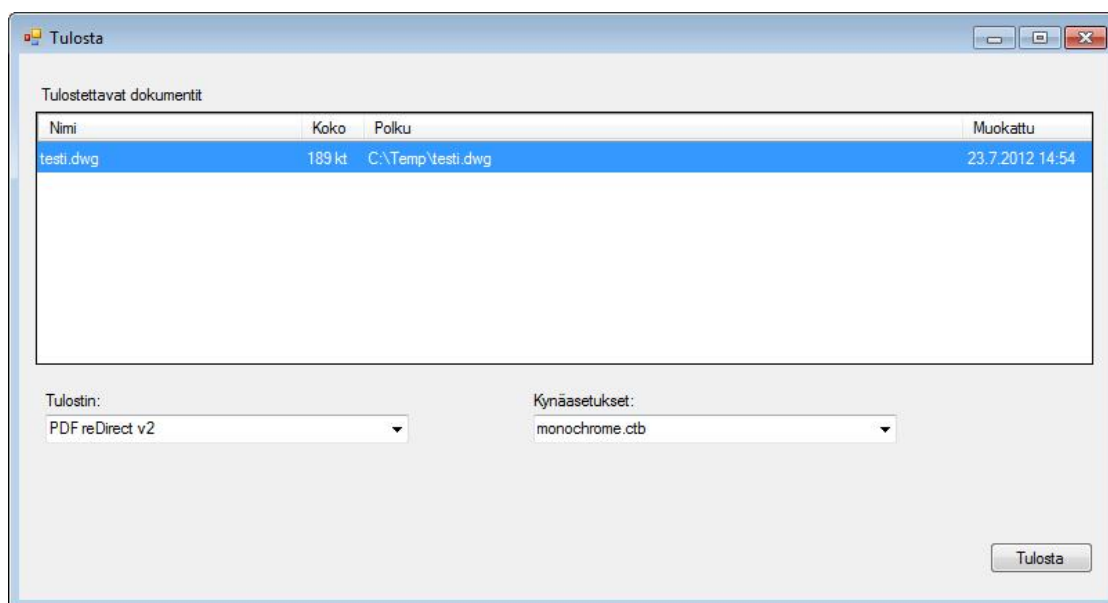
Yksi suunnittelujärjestelmän tärkeimmistä ominaisuuksista on tietojen hyödyntäminen useammassa kohteessa. Tietokannasta voidaan hakea halutut tiedot ja luoda halutunlainen raportti. ”Raportit esittävät yleisesti yhteenvedon laajemmasta tietosisällöstä.” /17, s. 219/. Raporttien avulla voidaan tietoja esittää halutulla tavalla, kuten laskemalla arvoja yhteen ja ryhmittämällä tietoja /5, s. 60/. Raportit voidaan luoda SQL-kyselyiden päälle.

Aluksi päätin, että luodaan kolme raporttia: kaapeli-, koje- ja kenttälaiteluettelo, joista kaapeliluettelo voidaan nähdä liitteessä 1. Raporttien tulostamisen tuli olla mahdollisimman yksinkertaista käyttäjälle. Sähkösuunnittelussa raportit ovat usein keskus- tai lähtökohtaisia. Luodut raportit ovat piirikohtaisia, eli valitaan piirit, joista halutaan raportti tulostaa. Piirien valitseminen tapahtuu valitsemalla halutut piiri –objektit puunäkymästä, jonka jälkeen valitaan yläriviltä haluttu raportti. Raportit avautuvat erillisiin ikkunoihin, mistä ne voidaan tulostaa tai tallentaa.

6.5 Tiedostojen tulostaminen

Elsan avulla voidaan myös tulostaa .dwg –muodossa olevia kuvia. dwg –kuvien tulostaminen on manuaalisesti hidasta, koska tarvitsee avata kuva ja tulostaa jokainen sivu erikseen.

Aluksi valitaan piiri–objektit puunäkymästä ja painetaan ”tulosta”, jolloin piirien mahdolliset dokumenttipolut viedään erilliselle ikkunalle. Tulostus–ikkunasta käyttäjä voi halutessaan poistaa dokumentit tulostettavien listasta ja valita tulostusasetukset.



KUVA 10. Tulostus-ikkuna

Tulostaminen tapahtuu AutoCAD 2013 mukana tulevan accoreconsole.exe:n avulla. Aluksi luodaan .bat –tiedosto, sekä .lsp –tiedosto.

Ladattaessa .bat –tiedosto ajaa Windows tiedoston sisältämät komennot komentorivillä. Tulostettaessa tiedoston sisältö on:

```
cd "C:\Program Files\Autodesk\AutoCAD 2013"
c:
cls
accoreconsole.exe /i "C:\Temp\testi.dwg" /s "C:\Program Files\Elsa\tulosta.scr" /l en-US
```

Ensimmäisellä kolmella rivillä navigoidaan hakemistoon, missä accoreconsole.exe –sovellus sijaitsee. Rivillä 4 ajetaan käsky accoreconsole.exessä. Rivien määrä riippuu kuvien määrästä.

”/i”, käytetään määrittämään piirustuksen tiedostopolku, missä script –tiedosto halutaan ajaa.

”/s”, käytetään määrittämään polku script –tiedostoon.

”/l”, käytetään määrittämään kieli. /19./

Tällöin accoreconsole.exe avaa kuvan, ja ajaa script -tiedoston nimeltä ”tulosta.scr”, joka sisältää:

```
FILEDIA
```

```
0
```

```
(LOAD "C:/Program Files/Elsa/Viewplot.lsp")
```

```
VIEWPLOT
```

```
FILEDIA
```

```
1
```

Kyseisessä tiedostossa ladataan Viewplot.lsp –tiedosto ja accoreconsole.exe –sovellus ajaa sen. Viewplot.lsp –tiedoston sisältö riippuu tulostusasetuksista.

7 VERTAILU

Tässä kappaleessa vertaillaan Elsa -suunnittelujärjestelmää muihin suunnittelujärjestelmiin. Tarkastelu tehdään lähinnä taloudellisesta näkökulmasta.

7.1 Ohjelmiston kehittyminen

Ohjelma saadaan usein valmiiksi, toimivaksi kokonaisuudeksi. Ohjelmia tarjoavan yrityksen pysyäkseen markkinoilla täytyy kehittää sovellusta jatkuvasti, esimerkiksi lisäten uusia ominaisuuksia tai ratkaisten uusien käyttöjärjestelmien tuomia ongelmia.

Ohjelman toiminnassa esiintyvät viat yritetään korjata mahdollisimman pian, mutta suurimmissa yrityksissä päivityksiä julkaistaan vain tietyn väliajoin, jolloin vikojen korjaamiseen saattaa mennä viikkoja tai kuukausiakin.

Itse kehitetyssä ohjelmassa vikojen korjaaminen onnistuu heti, kun viat ilmenevät. Tällöin työntekoa häiritsevät viat eivät mahdollisesti viivästytä projektien valmistumista.

Valmiiksi luotuihin ohjelmien toimintaan on usein vaikea päästä vaikuttamaan, vaikka monien ohjelmien ulkoasua voikin muokata omiin tarpeisiin sopivaksi. Itse ohjelmaa kehitettäessä pääsee työntekijät ja itse tekijä vaikuttamaan ohjelman toimintaan eli ohjelma voidaan räätälöidä tarkasti yrityksen tarpeisiin. Mahdolliset uudet päivityksetkään ohjelmista eivät yleensä ole ilmaisia, eli uusista ominaisuuksista joudutaan maksamaan. Itse kehitettyyn sovellukseen voidaan ominaisuuksia lisätä jatkuvasti, eikä niistä tarvitse maksaa, paitsi tekijän työnä.

7.2 Taloudellinen näkökulma

Suunnittelutoimistoilla olisi usein kolme vaihtoehtoa: ostaa tai vuokrata se valmiina ohjelmistoja tarjoavalta yritykseltä, kehittää sovellus itse tai olla käyttämättä suunnittelujärjestelmää. Tässä kappaleessa tarkastellaan eri vaihtoehtojen tuomia hyötyjä ja kustannuksia.

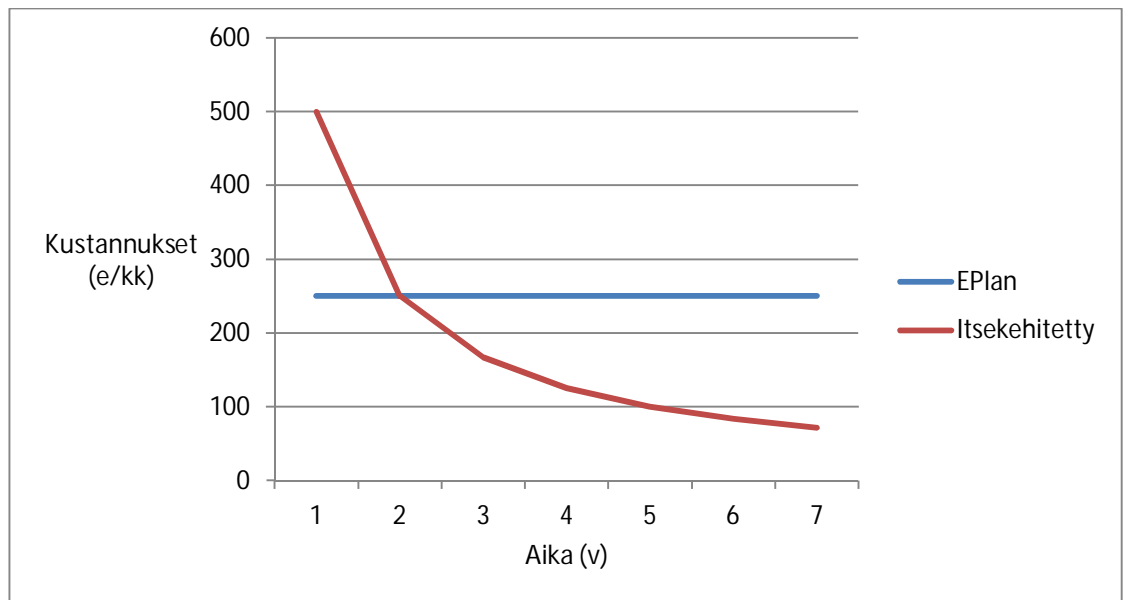
Suunnittelujärjestelmän vuokraaminen, esimerkiksi Eplanilta, maksaisi 250 euroa kuukaudessa per lisenssi. Vuokraamisen lisäksi kustannuksia syntyisi järjestelmän käyttöönotosta ja koulutuksesta.

Kouvolan Etteplanin toimipisteellä on noin 10 käyttäjää, joten kustannukseksi tulisi 2500 euroa kuukaudessa, jos suunnittelujärjestelmä vuokrattaisiin. Vuodessa kyseinen summa olisi 30 000 euroa.

Sovellusta itse kehitettäessä olisi tällöin käytössä vuodessa 30 000 euroa. Vuodessa ohjelmistokehitysalalla työskentelevä insinööri tienaa 3433e/kk /20, s. 9/. Oletetaan, että työntekijä maksaa yritykselle 1,5-kertaisesti, eli kehitystyöhön olisi käytettävissä noin 870 tuntia vuodessa. Sovelluksen kehittämiseen voitaisiin olettaa menevän noin 1700 tuntia, mutta tämän jälkeen sovelluksella ei olisi kustannuksia, paitsi kehitystarpeisiin tehty työ. Suunnittelujärjestelmän hinnaksi tulisi noin 60 000 euroa, eli se maksaisi itsensä takaisin Eplaniin verrattuna kahdessa vuodessa. Nykyisessä Etteplanin tilanteessa, koska suunnittelujärjestelmä on jo ennestään, on sen kehittäminen kaikista viisainta.

Käyttämättä suunnittelujärjestelmää kuluisi työtunteja piirustuksia kohti enemmän, mikä voisi johtaa markkinoilta putoamiseen, koska tuntiarviot tarjottavista projekteista kasvaisivat suuriksi. Vuodessa jäisi Eplaniin verrattuna noin 870 työtuntia enemmän, jos työntekijöitä olisi kymmenen, mikä olisi työntekijää kohden 87 tuntia.

Kuvasta 11 on nähtävissä kuukausittaiset kustannukset niiden jakamisesta eri aikaväleille. Kuvasta huomataan, että Eplanin kuukausittaiset kustannukset ovat kiinteät, mutta tarkasteluaikaa pidentämällä saadaan itse kehitetyn sovelluksen kuukausittaiset kustannukset pienemmiksi.



KUVA 11. Kuukausittaiset kustannukset ajan funktiona, kun käyttäjiä on 10

Itse kehitetyssä järjestelmässä käyttäjämäärä olisi käytännössä rajaton lähes samoilla kustannuksilla. Käyttäjämäärän kasvaessa jouduttaisiin siirtymään Jet –tietokannasta Microsoft SQL Serveriin, koska JET–tietokantomoottori ei ole kovin hyödyllinen, jos käyttäjiä on paljon /4, s. 14/.

8 POHDINTA

Opinnäytetyön tavoitteena oli miettiä mahdollisimman yksinkertainen ja tehokas tapa käsitellä valmiiksi suunnitellun tietokannan tietoja erillisen sovelluksen avulla sekä luoda sovellus, millä tehtävä onnistuisi. Sovelluksesta tuli sellainen kuin oli suunniteltu ja siitä löytyvät ne ominaisuudet, joita siltä vaadittiin. Opinnäytetyönä valmistunutta sovellusta ei ole vielä päästy käyttämään projekteissa, mutta sitä on päästy jo testaamaan, mikä on tärkeää mahdollisten virheiden löytymisen kannalta. Mahdolliset virheet on hyvä löytää mahdollisimman pian, jotta ne eivät hidasta projekteja. Tulevaisuudessa sovellusta tullaan käyttämään sekä kehittämään jatkuvasti.

Tietotekniikan hyödyntäminen suunnittelutyössä on yhä tärkeämmässä asemassa, kun kilpailevassa yhteiskunnassa työtä täytyy tehdä entistä tehokkaammin. Suunnittelujärjestelmien yleistyessä insinööritoimistoissa uskon työnantajien vaativan yhä parempaa tietämystä tietokannoista ja ohjelmoinnista. Pienilläkin ohjelmointipätkillä voi työntekijä tehostaa työn tuottavuuttansa huomattavasti.

Työ oli haastava, koska minulla ei ollut aiempaa ohjelmointikokemusta juuri lainkaan sekä vähäinen dokumentaatio joistakin aihealueista, kuten accoreconsolesta, tuotti vaikeuksia. Opinnäytetyön aihealue oli kattava, koska täytyi tutustua tiedonhallintaan ja sähkösuunnitteluun. Suuri hyöty ohjelmointiosuudessa oli internet, josta löytyi ohjelmoinnista monia oppaita ja esimerkkejä. Uskon, että opinnäytetyössäni oppimista taidoista on hyötyä minulle työelämässä tulevaisuudessakin, ja toivon, että opinnäytetyöstä olisi hyötyä myös muille.

LÄHTEET

- /1/ Etteplan Oyj. WWW-dokumentti. www.etteplan.fi. Luettu 20.11.2012. Päivitetty 20.11.2012.
- /2/ Etteplan Design Center Oy, Kouvola. WWW-dokumentti. www.lca.fi/alku.htm. Luettu 20.11.2012. Päivitetty 18.4.2011.
- /3/ Rantanen, Jaakko; Sainio, Arto; Laiho, Martti; Renkonen, Eero; Silpiö, Kari. Re-laatiotietokannat. Helsinki: Painatuskeskus Oy. 1993.
- /4/ Chung, Luke. Microsoft SQL Server 2005. Sähköinen dokumentti, 2006.
- /5/ Viescas, John. Running Microsoft Access 2000. Oy Edita Ab, 2000.
- /6/ Sähkö- ja teleurakoitsijaliitto 2004, Neuvottelevat sähkösuunnittelijat Ry. Sähkösuunnittelun käsikirja.
- /7/ Norelco Oy. WWW-dokumentti. <http://www.norelco.fi/index.php?pid=50&lg=fi>. Luettu 20.11.2012. Ei päivitystietoa.
- /8/ Suomen standardisoimisliitto. Pienjännitesähköasennukset ja sähkötyöturvallisuus. Helsinki: SFS, 2007.
- /9/ Microsoft. WWW-dokumentti. [http://msdn.microsoft.com/en-us/library/fx6bk1f4\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/fx6bk1f4(v=vs.71).aspx). Luettu 21.11.2012. Päivitetty 21.11.2012.
- /10/ Microsoft. WWW-dokumentti. [http://msdn.microsoft.com/en-us/library/czz35az4\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/czz35az4(v=vs.100).aspx). Luettu 21.11.2012. Päivitetty 21.11.2012.
- /11/ Microsoft. WWW-dokumentti. <http://www.microsoft.com/visualstudio/en-gb/products>. Luettu 21.11.2012. Päivitetty 21.11.2012.
- /12/ Microsoft. WWW-dokumentti. <http://www.microsoft.com/en-us/news/press/2006/apr06/04-19VSExpressFreePR.aspx>. Luettu 21.11.2012. Päivitetty 21.11.2012.
- /13/ Moghadampour, Ghodrat. C# Windows ja tietokanta-ohjelmointi. Jyväskylä: WSOYpro Oy. 2011.
- /14/ W3schools. WWW-dokumentti. http://www.w3schools.com/sql/sql_intro.asp. Luettu 21.11.2012. Päivitetty 21.11.2012.
- /15/ Autodesk, Inc. WWW-dokumentti. <http://usa.autodesk.com/company/>. Luettu 21.11.2012. Päivitetty 21.11.2012.
- /16/ AutoCAD. WWW-dokumentti. <http://usa.autodesk.com/autocad/>. Luettu 21.11.2012. Päivitetty 21.11.2012.
- /17/ Finkelstein, Ellen. AutoCAD 2012 & AutoCAD LT 2012 Bible. Indianapolis: Wiley Publishing, Inc., 2011.

/18/ Lambert, Steve; Lambert III, M. Dow; Preppernau, Joan. Access 2007 tehokas hallinta. Helsinki: Readme.fi. 2008.

/19/ Ramamoorthy, Balaji. AutoCAD DevBlog. WWW-dokumentti.
<http://adndevblog.typepad.com/autocad/2012/04/getting-started-with-accoreconsole.html>. Luettu 21.11.2012. Päivitetty 21.11.2012

/20/ Uusi Insinööriliitto UIL ry. 2011 Palkkatilastot, Insinöörien työmarkkinatutkimuksen TMT X-2011 tuloksia. Helsinki: Uusi Insinööriliitto ry, 2011.

LIITE 1.
Kaapeliluettelo

KAAPELILUETTELO							
Kaapelitunnus	Laji	Koko	Misä	Mihin	Pituus	Huom.	
Demopiri2-W01	MCMK	3x2,5+2,5	22N1-0102	-U1	10		
Demopiri2-W03	MCMK	3x2,5+2,5	-U1	-Q1	100		
Demopiri2-W02	MCMK	3x2,5+2,5	-Q1	-M1	5		
Demopiri2-W05	MMJ	2x1,5	22N1-0102	-Q1	110		
Demopiri1-W01	MCMK	3x2,5+2,5	22N1-0103	-U1	10		
Demopiri1-W03	MCMK	3x2,5+2,5	-U1	-Q1	100		
Demopiri1-W02	MCMK	3x2,5+2,5	-Q1	-M1	5		
Demopiri1-W05	MMJ	2x1,5	22N1-0103	22N1-0103	110		



Revisio: JILA
Määrä

Päivä: 23.11.2012

Proj.

Demopiri2, Demopiri1

Arvio

1

1